

AMENDMENTS TO THE CLAIMS

1. (Previously Presented) In a program development environment, a computer implemented method comprising:

providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

defining said assignment function in more than one class;

overloading said assignment function based on a context of said base language objects;

and

calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language.

2. (Canceled)

3. (Previously Presented) The method of claim 1 wherein said assignment function overloads a mathematical operator.

4. (Canceled)

5 (Previously Presented). The method of claim 1, further comprising:

evaluating said class at compile-time, and

adjusting said resulting class definitions from said evaluation to increase efficiency of run-time performance.

6. (Previously Presented) The method of claim 1, further comprising:

overloading a mathematical operator with said assignment function to alter a sequence of evaluation of operands usually followed in said programming language, said overloading designating an order of operand evaluation.

7. (Previously Presented) The method of claim 1, further comprising:

calling a method in said parse tree object to determine a type of operator at a root of said parse tree data structure.

8. (Previously Presented) The method of claim 1, further comprising:

calling a method in said parse tree object to retrieve one of an associated left and right tree.

9. (Previously Presented) The method of claim 1 wherein a root of said parse tree data structure is one of a constant, variable, a mathematical symbol and a mathematical expression.

10. (Original) The method of claim 1 wherein said assignment function is not explicitly defined.

11. (Original) The method of claim 1 wherein said assignment function is used to identify in-place operations.

12. (Original) The method of claim 1 wherein said assignment function is used to identify and perform multiply and accumulate ("MAC") operations.

13. (Original) The method of claim 1 wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework.

14. (Currently Amended) In a program development environment, a computer implemented method comprising:

providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a

class, said class being a basis for a plurality of parse tree objects, said parse tree objects including parse tree object methods that retrieve values for base language objects;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

defining said assignment function in more than one class;

overloading said assignment function based on a context of said base language objects, said context determined from one of said parse tree object methods;

calling said assignment function from said language processor to determine a value of at least one assignment within at least one of the said base language and a base language extension for said base language; and

generating code for an embedded processor using said parse tree data structure.

15. (Previously Presented) In a program development environment, a computer implemented method comprising:

providing, via a programming language, a language processor with built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

defining said assignment function in more than one class;

overloading said assignment function based on a context of said base language objects;

calling said assignment function from said language processor to determine a value of at least one assignment within at least one of the base language and a base language extension for said base language; and

using said parse tree data structure in software emulation.

16. (Previously Presented) A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

one or more instructions for providing a programming language;

one or more instructions for providing a language processor via the said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including parse tree object methods that retrieve values for base language objects;

one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

one or more instructions for defining said assignment function in more than one class;

one or more instructions for overloading said assignment function based on a context of said base language objects, said context determined from one of said parse tree object methods; and

one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language.

17. (Canceled)

18. (Previously Presented) The medium of claim 16 wherein said assignment function overloads a mathematical operator.

19. (Previously Presented) The medium of claim 16 wherein said overloading of said assignment function is based on a context of said base language objects.

20. (Previously Presented) The medium of claim 16, wherein said instructions further comprise:

one or more instructions for evaluating said class at compile-time, and

one or more instructions for adjusting said resulting class definitions from said evaluation to increase efficiency of run-time performance.

21. (Previously Presented) The medium of claim 16, wherein said instructions further comprise:

one or more instructions for overloading a mathematical operator with said assignment function to alter a sequence of evaluation of operands usually followed in said programming language, said overloading designating an order of operand evaluation.

22. (Previously Presented) The medium of claim 16, wherein said instructions further comprise: one or more instructions for calling a method in said parse tree object to determine a type of operator at a root of said parse tree data structure.

23. (Previously Presented) The medium of claim 16, wherein said instructions further comprise one or more instructions for calling a method in said parse tree object to retrieve one of an associated left and right tree.

24. (Original) The medium of claim 16 wherein a root of said parse tree data structure is one of a constant, variable, a mathematical symbol and a mathematical expression.

25. (Original) The medium of claim 16 wherein said assignment function is not explicitly defined.

26. (Original) The medium of claim 16 wherein said assignment function is used to identify in-place operations.

27. (Original) The medium of claim 16 wherein said assignment function is used to identify and perform multiply and accumulate ("MAC") operations.

28. (Original) The medium of claim 16 wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework.

29. (Original) The medium of claim 16 wherein said parse tree data structure is used to generate code for an embedded processor.

30. (Original) The medium of claim 16 wherein said parse tree data structure is used in processor emulation.

31-36. (Canceled)

37. (Previously Presented) A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

- one or more instructions for providing a programming language;

- one or more instructions for providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;

- one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

- one or more instructions for defining said assignment function in more than one class;

- one or more instructions for overloading said assignment function based on a context of said base language objects;

- one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language; and

- one or more instructions for generating code for an embedded processor using said parse tree data structure.

38. (Previously Presented) A computer-readable storage medium for storing computer executable instructions for use in a program development environment, said instructions comprising:

- one or more instructions for providing a programming language;

- one or more instructions for providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure written in a base language, said parse tree data structure represented as a class, said class being a basis for a plurality of parse tree objects, said parse tree objects including methods that retrieve values for base language objects;

one or more instructions for defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

one or more instructions for defining said assignment function in more than one class;

one or more instructions for overloading said assignment function based on a context of said base language objects;

one or more instructions for calling said assignment function from said language processor to determine a value of at least one assignment within at least one of said base language and a base language extension to said base language; and

one or more instructions for using said parse tree data structure in software emulation.

39. (Previously Presented) In an object-oriented program development environment having a base language, a computer implemented method comprising:

providing a programming language;

providing a language processor via said programming language, said language processor having built-in support for a parse tree data structure in said object-oriented program development environment, said parse tree data structure used as a basis for at least one parse tree object, said parse tree objects including methods that retrieve values for base language objects;

defining an assignment function taking a plurality of parse tree structures as arguments in more than one class;

calling said assignment function from said language processor to determine a value of at least one assignment within at least one of a base language and a base language extension; and

overloading a mathematical operator with said assignment function based on a context of a plurality of said base language objects.

40. (Previously Presented) A computer implemented method for providing a language processor in a program development environment, said method comprising:

building a parse tree data structure based on source code with the language processor;

instantiating a first parse tree object and a second parse tree object;

evaluating said second parse tree object to obtain a value, said evaluating done based on a context provided by said first parse tree object; and

assigning said value to said first parse tree object.